LWE Continued
&
Continuous LWE

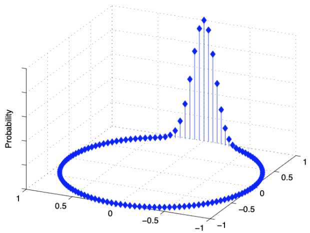
Yi Tang

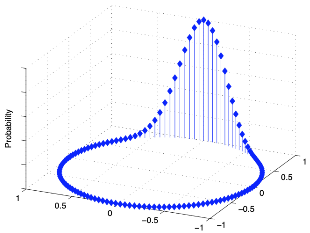
February 23, 2021
Last updated on June 5, 2021

Table of Contents

Recall: Definition of LWE

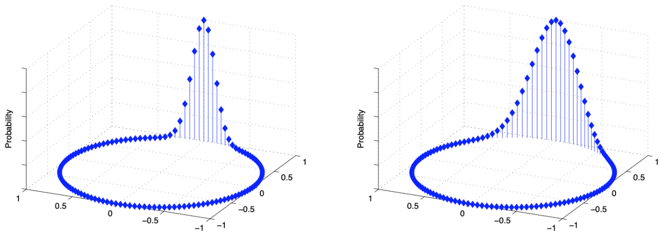
- Gaussian kernel: $\rho_s(\mathbf{x}) := \exp(-\pi\|\mathbf{x}/s\|^2)$ ($\sigma = s/\sqrt{2\pi}$)
- Gaussian distribution D_s: density ρ_s/s^n ($n = \dim \mathbf{x}$)
- Sample distribution $A_{\mathbf{s},\alpha}$ for $\mathbf{s} \in \mathbb{Z}_p^n$: $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s}\rangle + e_p)$ where $\mathbf{a} \sim \mathbb{Z}_p^n$ and $e_p = \lfloor pe \rceil \bmod p \in \mathbb{Z}_p$, $e \sim D_\alpha$
- Learning with errors $\mathrm{LWE}_{p,s}$:
  - Search: Given samples from $A_{\mathbf{s},\alpha}$, find $\mathbf{s}$
  - Decision: Distinguish between $A_{\mathbf{s},\alpha}$ and $U(\mathbb{Z}_p^{n+1})$





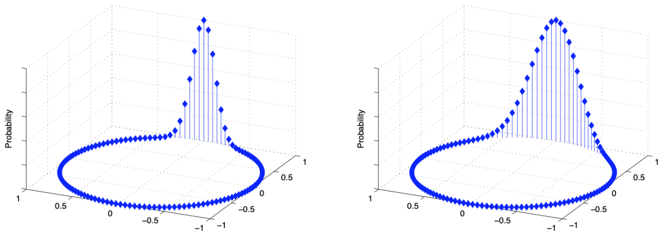
Recall: Definition of LWE

- Gaussian kernel: $\rho_s(\mathbf{x}) := \exp(-\pi\|\mathbf{x}/s\|^2)$ ($\sigma = s/\sqrt{2\pi}$)
- Gaussian distribution D_s: density ρ_s/s^n ($n = \dim \mathbf{x}$)
- Sample distribution $A_{\mathbf{s},\alpha}$ for $\mathbf{s} \in \mathbb{Z}_p^n$: $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s}\rangle + e_p)$ where $\mathbf{a} \sim \mathbb{Z}_p^n$ and $e_p = \lfloor pe \rceil \bmod p \in \mathbb{Z}_p$, $e \sim D_\alpha$
- Learning with errors $\mathrm{LWE}_{p,s}$:
  - Search: Given samples from $A_{\mathbf{s},\alpha}$, find $\mathbf{s}$
  - Decision: Distinguish between $A_{\mathbf{s},\alpha}$ and $U(\mathbb{Z}_p^{n+1})$



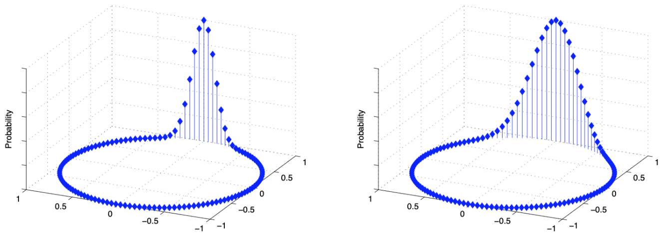
Recall: Definition of LWE

- Gaussian kernel: $\rho_s(\mathbf{x}) := \exp(-\pi\|\mathbf{x}/s\|^2)$ $(\sigma = s/\sqrt{2\pi})$
- Gaussian distribution D_s: density ρ_s/s^n $(n = \dim \mathbf{x})$
- Sample distribution $A_{\mathbf{s},\alpha}$ for $\mathbf{s} \in \mathbb{Z}_p^n$: $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s}\rangle + e_p)$ where $\mathbf{a} \sim \mathbb{Z}_p^n$ and $e_p = \lfloor pe \rceil \bmod p \in \mathbb{Z}_p$, $e \sim D_\alpha$
- Learning with errors $\mathrm{LWE}_{p,s}$:
  - Search: Given samples from $A_{\mathbf{s},\alpha}$, find $\mathbf{s}$
  - Decision: Distinguish between $A_{\mathbf{s},\alpha}$ and $U(\mathbb{Z}_p^{n+1})$



Recall: Definition of LWE

- Gaussian kernel: $\rho_s(\mathbf{x}) := \exp(-\pi\|\mathbf{x}/s\|^2)$ $(\sigma = s/\sqrt{2\pi})$
- Gaussian distribution D_s: density ρ_s/s^n $(n = \dim \mathbf{x})$
- Sample distribution $A_{\mathbf{s},\alpha}$ for $\mathbf{s} \in \mathbb{Z}_p^n$: $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s}\rangle + e_p)$ where $\mathbf{a} \sim \mathbb{Z}_p^n$ and $e_p = \lfloor pe \rceil \bmod p \in \mathbb{Z}_p$, $e \sim D_\alpha$
- Learning with errors $\mathrm{LWE}_{p,s}$:
  - Search: Given samples from $A_{\mathbf{s},\alpha}$, find $\mathbf{s}$
  - Decision: Distinguish between $A_{\mathbf{s},\alpha}$ and $U(\mathbb{Z}_p^{n+1})$



Variants of LWE

- Worst/average cases: whether $\mathbf{s} \in \mathbb{Z}_p^n$ is arbitrary or uniform
- "Continuous" variant of $A_{\mathbf{s},\alpha}$: $(\mathbf{a}, b = [\langle \mathbf{a}, \mathbf{s} \rangle / p + e] \bmod 1)$
- Reductions among variants:
  1. "Continuous" to discrete: discretize $b \in [0, 1)$ to $\lfloor pb \rceil \bmod p$
  2. Worst- to average-case: pick $\mathbf{t} \sim \mathbb{Z}_p^n$ and transform worst-case samples $(\mathbf{a}, b)$ to $(\mathbf{a}, b + \langle \mathbf{a}, \mathbf{t} \rangle) \sim A_{\mathbf{s}+\mathbf{t},\alpha}$
  3. Search to decision: transform LWE samples $(\mathbf{a}, b)$ to $(\mathbf{a} + \ell \mathbf{e}_i, b + \ell k)$ where $\ell \sim \mathbb{Z}_p$, which $\sim A_{\mathbf{s},\alpha}$ if $k = s_i$ and is uniform (requiring prime p) otherwise, and brute-force s_i (requiring poly p)
  4. Decision to search: search & verify
- As a result,
  1. Want reduction to "continuous", worst-case, decisional LWE
  2. Build applications on arbitrary (discrete, average-case) LWE

Variants of LWE

- Worst/average cases: whether $\mathbf{s} \in \mathbb{Z}_p^n$ is arbitrary or uniform
- "Continuous" variant of $A_{\mathbf{s},\alpha}$: $(\mathbf{a}, b = [\langle \mathbf{a}, \mathbf{s} \rangle / p + e] \bmod 1)$
- Reductions among variants:
  1. "Continuous" to discrete: discretize $b \in [0, 1)$ to $\lfloor pb \rceil \bmod p$
  2. Worst- to average-case: pick $\mathbf{t} \sim \mathbb{Z}_p^n$ and transform worst-case samples $(\mathbf{a}, b)$ to $(\mathbf{a}, b + \langle \mathbf{a}, \mathbf{t} \rangle) \sim A_{\mathbf{s}+\mathbf{t},\alpha}$
  3. Search to decision: transform LWE samples $(\mathbf{a}, b)$ to $(\mathbf{a} + \ell \mathbf{e}_i, b + \ell k)$ where $\ell \sim \mathbb{Z}_p$, which $\sim A_{\mathbf{s},\alpha}$ if $k = s_i$ and is uniform (requiring prime p) otherwise, and brute-force s_i (requiring poly p)
  4. Decision to search: search & verify
- As a result,
  1. Want reduction to "continuous", worst-case, decisional LWE
  2. Build applications on arbitrary (discrete, average-case) LWE

Variants of LWE

- Worst/average cases: whether $\mathbf{s} \in \mathbb{Z}_p^n$ is arbitrary or uniform
- "Continuous" variant of $A_{\mathbf{s},\alpha}$: $(\mathbf{a}, b = [\langle \mathbf{a}, \mathbf{s} \rangle / p + e] \bmod 1)$
- Reductions among variants:
  1. "Continuous" to discrete: discretize $b \in [0, 1)$ to $\lfloor pb \rceil \bmod p$
  2. Worst- to average-case: pick $\mathbf{t} \sim \mathbb{Z}_p^n$ and transform worst-case samples $(\mathbf{a}, b)$ to $(\mathbf{a}, b + \langle \mathbf{a}, \mathbf{t} \rangle) \sim A_{\mathbf{s}+\mathbf{t},\alpha}$
  3. Search to decision: transform LWE samples $(\mathbf{a}, b)$ to $(\mathbf{a} + \ell \mathbf{e}_i, b + \ell k)$ where $\ell \sim \mathbb{Z}_p$, which $\sim A_{\mathbf{s},\alpha}$ if $k = s_i$ and is uniform (requiring prime p) otherwise, and brute-force s_i (requiring poly p)
  4. Decision to search: search & verify
- As a result,
  1. Want reduction to "continuous", worst-case, decisional LWE
  2. Build applications on arbitrary (discrete, average-case) LWE

Variants of LWE

- Worst/average cases: whether $\mathbf{s} \in \mathbb{Z}_p^n$ is arbitrary or uniform
- "Continuous" variant of $A_{\mathbf{s},\alpha}$: $(\mathbf{a}, b = [\langle \mathbf{a}, \mathbf{s} \rangle / p + e] \bmod 1)$
- Reductions among variants:
  1. "Continuous" to discrete: discretize $b \in [0, 1)$ to $\lfloor pb \rceil \bmod p$
  2. Worst- to average-case: pick $\mathbf{t} \sim \mathbb{Z}_p^n$ and transform worst-case samples $(\mathbf{a}, b)$ to $(\mathbf{a}, b + \langle \mathbf{a}, \mathbf{t} \rangle) \sim A_{\mathbf{s}+\mathbf{t},\alpha}$
  3. Search to decision: transform LWE samples $(\mathbf{a}, b)$ to $(\mathbf{a} + \ell \mathbf{e}_i, b + \ell k)$ where $\ell \sim \mathbb{Z}_p$, which $\sim A_{\mathbf{s},\alpha}$ if $k = s_i$ and is uniform (requiring prime p) otherwise, and brute-force s_i (requiring poly p)
  4. Decision to search: search & verify
- As a result,
  1. Want reduction to "continuous", worst-case, decisional LWE
  2. Build applications on arbitrary (discrete, average-case) LWE

Variants of LWE

- Worst/average cases: whether $\mathbf{s} \in \mathbb{Z}_p^n$ is arbitrary or uniform
- "Continuous" variant of $A_{\mathbf{s},\alpha}$: $(\mathbf{a}, b = [\langle \mathbf{a}, \mathbf{s} \rangle / p + e] \bmod 1)$
- Reductions among variants:
  1. "Continuous" to discrete: discretize $b \in [0, 1)$ to $\lfloor pb \rceil \bmod p$
  2. Worst- to average-case: pick $\mathbf{t} \sim \mathbb{Z}_p^n$ and transform worst-case samples $(\mathbf{a}, b)$ to $(\mathbf{a}, b + \langle \mathbf{a}, \mathbf{t} \rangle) \sim A_{\mathbf{s}+\mathbf{t},\alpha}$
  3. Search to decision: transform LWE samples $(\mathbf{a}, b)$ to $(\mathbf{a} + \ell \mathbf{e}_i, b + \ell k)$ where $\ell \sim \mathbb{Z}_p$, which $\sim A_{\mathbf{s},\alpha}$ if $k = s_i$ and is uniform (requiring prime p) otherwise, and brute-force s_i (requiring poly p)
  4. Decision to search: search & verify
- As a result,
  1. Want reduction to "continuous", worst-case, decisional LWE
  2. Build applications on arbitrary (discrete, average-case) LWE

Variants of LWE

- Worst/average cases: whether $\mathbf{s} \in \mathbb{Z}_p^n$ is arbitrary or uniform
- "Continuous" variant of $A_{\mathbf{s},\alpha}$: $(\mathbf{a}, b = [\langle \mathbf{a}, \mathbf{s} \rangle / p + e] \bmod 1)$
- Reductions among variants:
  1. "Continuous" to discrete: discretize $b \in [0,1)$ to $\lfloor pb \rceil \bmod p$
  2. Worst- to average-case: pick $\mathbf{t} \sim \mathbb{Z}_p^n$ and transform worst-case samples $(\mathbf{a}, b)$ to $(\mathbf{a}, b + \langle \mathbf{a}, \mathbf{t} \rangle) \sim A_{\mathbf{s}+\mathbf{t},\alpha}$
  3. Search to decision: transform LWE samples $(\mathbf{a}, b)$ to $(\mathbf{a} + \ell \mathbf{e}_i, b + \ell k)$ where $\ell \sim \mathbb{Z}_p$, which $\sim A_{\mathbf{s},\alpha}$ if $k = s_i$ and is uniform (requiring prime p) otherwise, and brute-force s_i (requiring poly p)
  4. Decision to search: search & verify
- As a result,
  1. Want reduction to "continuous", worst-case, decisional LWE
  2. Build applications on arbitrary (discrete, average-case) LWE

Lattice Problems

- Shortest vector problem SVP_γ: Given rank-n lattice $\mathcal{L}$, find nonzero $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v}\| \leq \gamma(n) \cdot \lambda_1(\mathcal{L})$
- GapSVP_γ: Given rank-n lattice $\mathcal{L}$ and length d, distinguish between YES: $\lambda_1(\mathcal{L}) \leq d$ and NO: $\lambda_1(\mathcal{L}) > \gamma(n) \cdot d$
- Shortest independent vectors problem SIVP_γ: Given rank-n lattice $\mathcal{L}$, find linearly independent $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathcal{L}$ such that $\max_i \|\mathbf{v}_i\| \leq \gamma(n) \cdot \lambda_n(\mathcal{L})$ [1]

[1] λ_n: $\lambda_n(\mathcal{L}) = \min\{\max_i \|\mathbf{v}_i\| : \mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathcal{L} \text{ and lin. ind.}\}$, "successive minima"

Lattice Problems

- Shortest vector problem SVP_γ: Given rank-n lattice $\mathcal{L}$, find nonzero $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v}\| \leq \gamma(n) \cdot \lambda_1(\mathcal{L})$
- GapSVP_γ: Given rank-n lattice $\mathcal{L}$ and length d, distinguish between YES: $\lambda_1(\mathcal{L}) \leq d$ and NO: $\lambda_1(\mathcal{L}) > \gamma(n) \cdot d$
- Shortest independent vectors problem SIVP_γ: Given rank-n lattice $\mathcal{L}$, find linearly independent $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathcal{L}$ such that $\max_i \|\mathbf{v}_i\| \leq \gamma(n) \cdot \lambda_n(\mathcal{L})$ [1]

---

[1] λ_n: $\lambda_n(\mathcal{L}) = \min\{\max_i \|\mathbf{v}_i\| : \mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathcal{L} \text{ and lin. ind.}\}$, "successive minima"

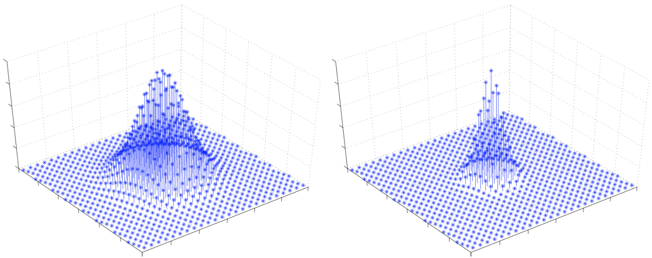
Lattice Problems

- Shortest vector problem SVP_γ: Given rank-n lattice $\mathcal{L}$, find nonzero $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v}\| \leq \gamma(n) \cdot \lambda_1(\mathcal{L})$
- GapSVP_γ: Given rank-n lattice $\mathcal{L}$ and length d, distinguish between YES: $\lambda_1(\mathcal{L}) \leq d$ and NO: $\lambda_1(\mathcal{L}) > \gamma(n) \cdot d$
- Shortest independent vectors problem SIVP_γ: Given rank-n lattice $\mathcal{L}$, find linearly independent $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathcal{L}$ such that $\max_i \|\mathbf{v}_i\| \leq \gamma(n) \cdot \lambda_n(\mathcal{L})$ [1]

[1] λ_n: $\lambda_n(\mathcal{L}) = \min\{\max_i \|\mathbf{v}_i\| : \mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathcal{L} \text{ and lin. ind.}\}$, "successive minima"

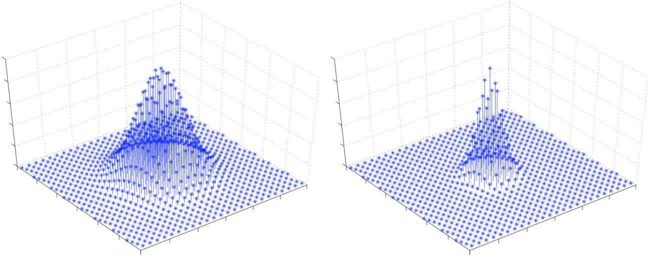
Lattice Problems: Discrete Gaussian

- Discrete Gaussian distribution $D_{\mathcal{L},s}$: over $\mathcal{L}$, probability distribution $\rho_s(\mathbf{v}) / \sum_{\mathbf{v}\in\mathcal{L}} \rho_s(\mathbf{v})$
- Discrete Gaussian sampling DGS_{φ}: Given rank-n lattice $\mathcal{L}$ and *width* $r \geq \varphi(\mathcal{L})$, sample with distribution $\overset{s}{\approx} D_{\mathcal{L},r}$



Lattice Problems: Discrete Gaussian

- Discrete Gaussian distribution $D_{\mathcal{L},s}$: over $\mathcal{L}$, probability distribution $\rho_s(\mathbf{v})/\sum_{\mathbf{v}\in\mathcal{L}}\rho_s(\mathbf{v})$
- Discrete Gaussian sampling DGS_{φ}: Given rank-n lattice $\mathcal{L}$ and *width* $r \geq \varphi(\mathcal{L})$, sample with distribution $\overset{s}{\approx} D_{\mathcal{L},r}$



Reduction from Lattice Problems to DGS

Results:

- $\mathrm{GapSVP}_{100\sqrt{n}\gamma(n)}$ to $\mathrm{DGS}_{\sqrt{n}\gamma(n)/\lambda_1(\mathcal{L}^*)}$
- $\mathrm{SIVP}_{2\sqrt{n}\gamma(n)}$ to $\mathrm{DGS}_{\gamma(n)\lambda_n(\mathcal{L})}$, for large enough γ (in particular $\gamma(n)\lambda_n(\mathcal{L}) \geq \sqrt{2}\,\eta_\varepsilon(\mathcal{L})$,[2] $\varepsilon \leq 1/10$)

Setting parameters:

- Will use $\mathrm{DGS}_{\sqrt{2n}\,\eta_\varepsilon(\mathcal{L})/\alpha}$ (for negligible ε), corresponding to
- $\mathrm{GapSVP}_{O(n/\alpha)}$ ($\eta_\varepsilon(\mathcal{L})\,\lambda_1(\mathcal{L}^*) \leq \sqrt{n}$ for $\varepsilon = 2^{-n}$)
- $\mathrm{SIVP}_{\tilde{O}(n/\alpha)}$ ($\eta_\varepsilon(\mathcal{L})/\lambda_n(\mathcal{L}) \leq \mathrm{polylog}(n)$)

[2] η_ε: "smoothing parameter", beyond which the discrete Gaussian "behaves like" continuous Gaussian with the same width

Reduction from Lattice Problems to DGS

Results:

- $\mathrm{GapSVP}_{100\sqrt{n}\gamma(n)}$ to $\mathrm{DGS}_{\sqrt{n}\gamma(n)/\lambda_1(\mathcal{L}^*)}$
- $\mathrm{SIVP}_{2\sqrt{n}\gamma(n)}$ to $\mathrm{DGS}_{\gamma(n)\,\lambda_n(\mathcal{L})}$, for large enough γ (in particular $\gamma(n)\,\lambda_n(\mathcal{L}) \geq \sqrt{2}\,\eta_\varepsilon(\mathcal{L})$,[2] $\varepsilon \leq 1/10$)

Setting parameters:

- Will use $\mathrm{DGS}_{\sqrt{2n}\,\eta_\varepsilon(\mathcal{L})/\alpha}$ (for negligible ε), corresponding to
- $\mathrm{GapSVP}_{O(n/\alpha)}$ ($\eta_\varepsilon(\mathcal{L})\,\lambda_1(\mathcal{L}^*) \leq \sqrt{n}$ for $\varepsilon = 2^{-n}$)
- $\mathrm{SIVP}_{\tilde{O}(n/\alpha)}$ ($\eta_\varepsilon(\mathcal{L})/\lambda_n(\mathcal{L}) \leq \mathrm{polylog}(n)$)

[2] η_ε: "smoothing parameter", beyond which the discrete Gaussian "behaves like" continuous Gaussian with the same width

Reduction from Lattice Problems to DGS

Results:

- $\text{GapSVP}_{100\sqrt{n}\gamma(n)}$ to $\text{DGS}_{\sqrt{n}\gamma(n)/\lambda_1(\mathcal{L}^*)}$
- $\text{SIVP}_{2\sqrt{n}\gamma(n)}$ to $\text{DGS}_{\gamma(n)\lambda_n(\mathcal{L})}$, for large enough γ (in particular $\gamma(n)\lambda_n(\mathcal{L}) \geq \sqrt{2}\,\eta_\varepsilon(\mathcal{L})$,[2] $\varepsilon \leq 1/10$)

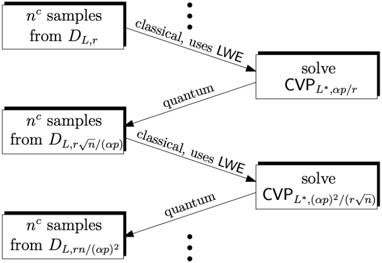
Setting parameters:

- Will use $\text{DGS}_{\sqrt{2n}\,\eta_\varepsilon(\mathcal{L})/\alpha}$ (for negligible ε), corresponding to
- $\text{GapSVP}_{O(n/\alpha)}$ ($\eta_\varepsilon(\mathcal{L})\,\lambda_1(\mathcal{L}^*) \leq \sqrt{n}$ for $\varepsilon = 2^{-n}$)
- $\text{SIVP}_{\widetilde{O}(n/\alpha)}$ ($\eta_\varepsilon(\mathcal{L})/\lambda_n(\mathcal{L}) \leq \text{polylog}(n)$)

[2] η_ε: "smoothing parameter", beyond which the discrete Gaussian "behaves like" continuous Gaussian with the same width

Reduction from DGS to LWE: Overview

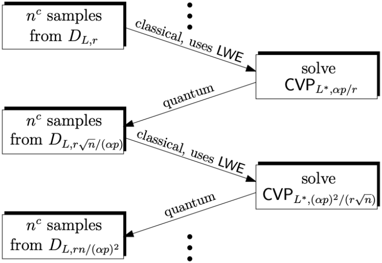
- Bootstrapping: $\mathsf{DGS}_{2^{2n}\lambda_n(\mathcal{L})}$ is efficiently sampleable (LLL-reduce, sample from continuous, and round)
- Iteratively "refine" the samples, using $\mathsf{LWE}_{p,\alpha}$ oracle, via intermediate problem BDD
- Finally reach desired $\mathsf{DGS}_{\sqrt{2n}\,\eta_\varepsilon(\mathcal{L})/\alpha}$ $(\eta_\varepsilon / \lambda_n = \Omega(1/n))$



(modern name for the specific CVP is BDD; omitting some $\sqrt{2}$ factors)

Reduction from DGS to LWE: Overview

- Bootstrapping: $\text{DGS}_{2^{2n}\lambda_n(\mathcal{L})}$ is efficiently sampleable (LLL-reduce, sample from continuous, and round)
- Iteratively "refine" the samples, using $\text{LWE}_{p,\alpha}$ oracle, via intermediate problem BDD
- Finally reach desired $\text{DGS}_{\sqrt{2n}\,\eta_\varepsilon(\mathcal{L})/\alpha}$ $(\eta_\varepsilon / \lambda_n = \Omega(1/n))$



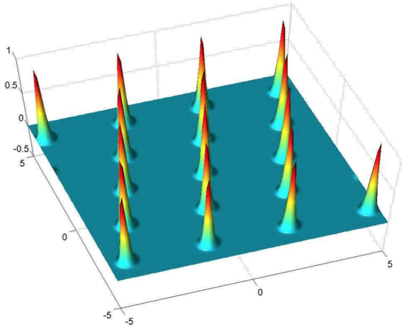
(modern name for the specific CVP is BDD; omitting some $\sqrt{2}$ factors)

Reduction from DGS to LWE: BDD

- (Closest vector problem CVP_γ: Given rank-n lattice $\mathcal{L}$ and target $\mathbf{t}$, find $\mathbf{v} \in \mathcal{L}$ such that $\mathrm{dist}(\mathbf{t}, \mathbf{v}) \leq \gamma(n) \cdot \mathrm{dist}(\mathbf{t}, \mathcal{L})$)
- Bounded distance decoding BDD_φ: Given rank-n lattice $\mathcal{L}$ and target $\mathbf{t}$ *satisfying* $\mathrm{dist}(\mathbf{t}, \mathcal{L}) \leq \varphi(\mathcal{L})$, find $\mathbf{v} \in \mathcal{L}$ such that $\mathrm{dist}(\mathbf{t}, \mathbf{v}) = \mathrm{dist}(\mathbf{t}, \mathcal{L})$

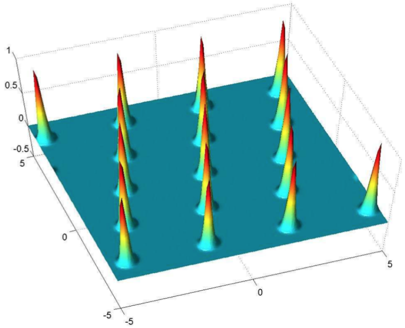
Why BDD?

- How to sample from DG if *with quantum*?
- Fourier transform of DG: $\widehat{D}_{\mathcal{L},r} \approx \exp(-\pi(r \cdot \mathrm{dist}(\mathbf{t}, \mathcal{L}^*))^2)$
- Easy to "compute": $\mathbf{t} = \mathbf{u} + \mathbf{t}'$, where $\mathbf{u} \sim \mathcal{L}^*, \mathbf{t}' \sim D^n_{1/r}$
- While quantum requires reversibility / "uncomputing";
  i.e. given $\mathbf{t} = \mathbf{u} + \mathbf{t}'$, find $\mathbf{u}$ / find $\mathbf{t}'$;
  i.e. BDD!



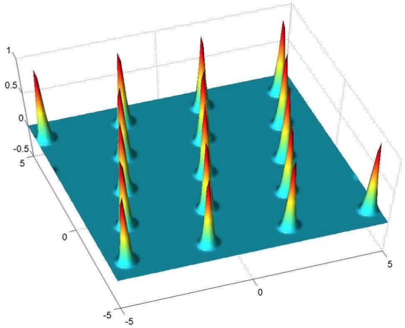
Why BDD?

- How to sample from DG if *with quantum*?
- Fourier transform of DG: $\widehat{D}_{\mathcal{L},r} \approx \exp(-\pi(r \cdot \mathrm{dist}(\mathbf{t}, \mathcal{L}^*))^2)$
- Easy to "compute": $\mathbf{t} = \mathbf{u} + \mathbf{t}'$, where $\mathbf{u} \sim \mathcal{L}^*, \mathbf{t}' \sim D^n_{1/r}$
- While quantum requires reversibility / "uncomputing";
  i.e. given $\mathbf{t} = \mathbf{u} + \mathbf{t}'$, find $\mathbf{u}$ / find $\mathbf{t}'$;
  i.e. BDD!



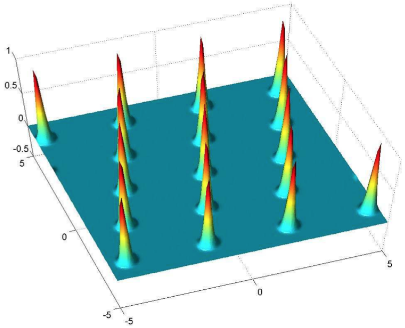
Why BDD?

- How to sample from DG if *with quantum*?
- Fourier transform of DG: $\widehat{D}_{\mathcal{L},r} \approx \exp(-\pi(r \cdot \mathrm{dist}(\mathbf{t}, \mathcal{L}^*))^2)$
- Easy to "compute": $\mathbf{t} = \mathbf{u} + \mathbf{t}'$, where $\mathbf{u} \sim \mathcal{L}^*, \mathbf{t}' \sim D^n_{1/r}$
- While quantum requires reversibility / "uncomputing"; i.e. given $\mathbf{t} = \mathbf{u} + \mathbf{t}'$, find $\mathbf{u}$ / find $\mathbf{t}'$; i.e. BDD!



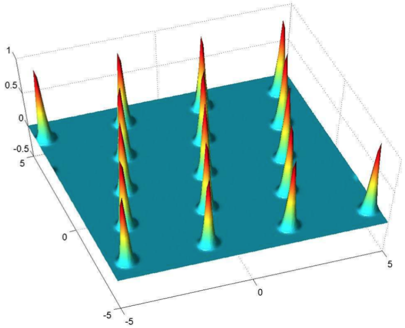
Why BDD?

- How to sample from DG if *with quantum*?
- Fourier transform of DG: $\widehat{D}_{\mathcal{L},r} \approx \exp(-\pi(r \cdot \mathrm{dist}(\mathbf{t}, \mathcal{L}^*))^2)$
- Easy to "compute": $\mathbf{t} = \mathbf{u} + \mathbf{t}'$, where $\mathbf{u} \sim \mathcal{L}^*, \mathbf{t}' \sim D^n_{1/r}$
- While quantum requires reversibility / "uncomputing";
  i.e. given $\mathbf{t} = \mathbf{u} + \mathbf{t}'$, find $\mathbf{u}$ / find $\mathbf{t}'$;
  i.e. BDD!



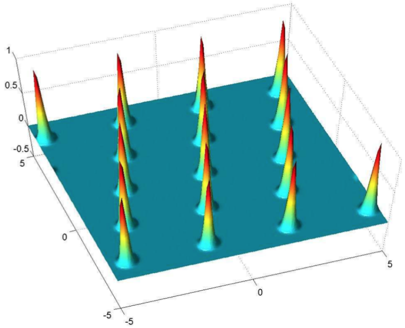
Why BDD?

- How to sample from DG if *with quantum*?
- Fourier transform of DG: $\widehat{D}_{\mathcal{L},r} \approx \exp(-\pi(r \cdot \mathrm{dist}(\mathbf{t}, \mathcal{L}^*))^2)$
- Easy to "compute": $\mathbf{t} = \mathbf{u} + \mathbf{t}'$, where $\mathbf{u} \sim \mathcal{L}^*, \mathbf{t}' \sim D^n_{1/r}$
- While quantum requires reversibility / "uncomputing";
  i.e. given $\mathbf{t} = \mathbf{u} + \mathbf{t}'$, find $\mathbf{u}$ / find $\mathbf{t}'$;
  i.e. BDD!



Why BDD?

- How to sample from DG if *with quantum*?
- Fourier transform of DG: $\widehat{D}_{\mathcal{L},r} \approx \exp(-\pi(r \cdot \mathrm{dist}(\mathbf{t}, \mathcal{L}^*))^2)$
- Easy to "compute": $\mathbf{t} = \mathbf{u} + \mathbf{t}'$, where $\mathbf{u} \sim \mathcal{L}^*, \mathbf{t}' \sim D^n_{1/r}$
- While quantum requires reversibility / "uncomputing";
  i.e. given $\mathbf{t} = \mathbf{u} + \mathbf{t}'$, find $\mathbf{u}$ / find $\mathbf{t}'$;
  i.e. BDD!



Reduction from DGS to BDD

$\mathrm{DGS}_{\sqrt{n}/(\sqrt{2}\varphi(\mathcal{L}^*))}$ to BDD with bound $\varphi(\mathcal{L}^*)$, for small enough φ (in particular $\varphi < \lambda_1/2$, *unique decoding*)

- Using quantum
- DGS on $\mathcal{L}$ always reduced to BDD on $\mathcal{L}^*$ (writing "with bound" instead of BDD_φ for less ambiguity)

Next step:

- Want to reduce BDD to DGS with larger width
- However can only do so with roughly width $1/\varphi(\mathcal{L}^*)$
- LWE for help!

Reduction from DGS to BDD

$\mathrm{DGS}_{\sqrt{n}/(\sqrt{2}\varphi(\mathcal{L}^*))}$ to BDD with bound $\varphi(\mathcal{L}^*)$, for small enough φ (in particular $\varphi < \lambda_1/2$, *unique decoding*)

- Using quantum
- DGS on $\mathcal{L}$ always reduced to BDD on $\mathcal{L}^*$ (writing "with bound" instead of BDD_φ for less ambiguity)

Next step:

- Want to reduce BDD to DGS with larger width
- However can only do so with roughly width $1/\varphi(\mathcal{L}^*)$
- LWE for help!

Reduction from DGS to BDD

$\text{DGS}_{\sqrt{n}/(\sqrt{2}\varphi(\mathcal{L}^*))}$ to BDD with bound $\varphi(\mathcal{L}^*)$, for small enough φ (in particular $\varphi < \lambda_1/2$, *unique decoding*)

- Using quantum
- DGS on $\mathcal{L}$ always reduced to BDD on $\mathcal{L}^*$
  (writing "with bound" instead of BDD_φ for less ambiguity)

Next step:

- Want to reduce BDD to DGS with larger width
- However can only do so with roughly width $1/\varphi(\mathcal{L}^*)$
- LWE for help!

Reduction from DGS to BDD

$\mathrm{DGS}_{\sqrt{n}/(\sqrt{2}\varphi(\mathcal{L}^*))}$ to BDD with bound $\varphi(\mathcal{L}^*)$, for small enough φ (in particular $\varphi < \lambda_1/2$, *unique decoding*)

- Using quantum
- DGS on $\mathcal{L}$ always reduced to BDD on $\mathcal{L}^*$ (writing "with bound" instead of BDD_φ for less ambiguity)

Next step:

- Want to reduce BDD to DGS with larger width
- However can only do so with roughly width $1/\varphi(\mathcal{L}^*)$
- LWE for help!

Reduction from DGS to BDD

$\mathrm{DGS}_{\sqrt{n}/(\sqrt{2}\varphi(\mathcal{L}^*))}$ to BDD with bound $\varphi(\mathcal{L}^*)$, for small enough φ (in particular $\varphi < \lambda_1/2$, *unique decoding*)

- Using quantum
- DGS on $\mathcal{L}$ always reduced to BDD on $\mathcal{L}^*$
  (writing "with bound" instead of BDD_φ for less ambiguity)

Next step:

- Want to reduce BDD to DGS with larger width
- However can only do so with roughly width $1/\varphi(\mathcal{L}^*)$
- LWE for help!

Reduction from BDD to DGS + (Search) LWE

BDD *with bound* $\alpha p/\sqrt{2}r$ to DGS_r + search $\mathrm{LWE}_{p,\alpha}$, for large enough r (in particular $r > \sqrt{2}p\,\eta_\varepsilon(\mathcal{L})$)

- Given $D_{\mathcal{L},r}$ samples (*instead of oracle*) and LWE oracle
- Want to solve BDD instance $(\mathcal{L}^*, \mathbf{t})$
  (remark: the $\mathcal{L}^*$ part is always the same)
- Transform DG samples $\mathbf{v}$ to LWE samples
  $(\mathcal{L}^{-1}\mathbf{v} \bmod p, [\langle \mathbf{v}, \mathbf{t}\rangle/p + e] \bmod 1)$ where $e \sim D_{\alpha/\sqrt{2}}$
  - Suppose $\mathbf{v} = \mathcal{L}(\mathbf{a} + p\mathbf{a}')$, $\mathbf{a} \in \mathbb{Z}_p^n$, and $\mathbf{t} = \mathcal{L}^*\mathbf{s} + \mathbf{t}'$, $\mathcal{L}^*\mathbf{s}$ is CV
  - $\langle \mathcal{L}(p\mathbf{a}'), \mathcal{L}^*\mathbf{s}\rangle/p \bmod 1 = 0$
  - $\langle \mathbf{v}, \mathbf{t}'\rangle/p \overset{s}{\approx}$ Gaussian noise of width $r\|\mathbf{t}'\|/p \leq \alpha/\sqrt{2}$
    (large enough r, "behaves like" continuous Gaussian!)
  - Samples $\overset{s}{\approx} (\mathbf{a}, [\langle \mathbf{a}, \mathbf{s}\rangle/p + e'] \bmod 1)$ where $e' \sim D_{\leq\alpha}$, $\mathbf{a} \overset{s}{\approx} U$
- Find $\mathbf{s}_0 = \mathbf{s} \bmod p$ using (enhanced) LWE oracle
- Recurse with $(\mathbf{t} - \mathcal{L}^*\mathbf{s}_0)/p$ for next base-p digits of $\mathbf{s}$
  (this reduces $\|\mathbf{t}'\|$ by p so finally can apply trivial algorithm)

Reduction from BDD to DGS + (Search) LWE

BDD *with bound* $\alpha p/\sqrt{2}r$ to DGS_r + search $\mathsf{LWE}_{p,\alpha}$, for large enough r (in particular $r > \sqrt{2}p\,\eta_\varepsilon(\mathcal{L})$)

- Given $D_{\mathcal{L},r}$ samples (*instead of oracle*) and LWE oracle
- Want to solve BDD instance $(\mathcal{L}^*, \mathbf{t})$
  (remark: the $\mathcal{L}^*$ part is always the same)
- Transform DG samples $\mathbf{v}$ to LWE samples
  $(\mathcal{L}^{-1}\mathbf{v} \bmod p, [\langle \mathbf{v}, \mathbf{t}\rangle/p + e] \bmod 1)$ where $e \sim D_{\alpha/\sqrt{2}}$
  - Suppose $\mathbf{v} = \mathcal{L}(\mathbf{a} + p\mathbf{a}')$, $\mathbf{a} \in \mathbb{Z}_p^n$, and $\mathbf{t} = \mathcal{L}^*\mathbf{s} + \mathbf{t}'$, $\mathcal{L}^*\mathbf{s}$ is CV
  - $\langle \mathcal{L}(p\mathbf{a}'), \mathcal{L}^*\mathbf{s}\rangle/p \bmod 1 = 0$
  - $\langle \mathbf{v}, \mathbf{t}'\rangle/p \overset{s}{\approx}$ Gaussian noise of width $r\|\mathbf{t}'\|/p \leq \alpha/\sqrt{2}$
    (large enough r, "behaves like" continuous Gaussian!)
  - Samples $\overset{s}{\approx} (\mathbf{a}, [\langle \mathbf{a}, \mathbf{s}\rangle/p + e'] \bmod 1)$ where $e' \sim D_{\leq\alpha}$, $\mathbf{a} \overset{s}{\approx} U$
- Find $\mathbf{s}_0 = \mathbf{s} \bmod p$ using (enhanced) LWE oracle
- Recurse with $(\mathbf{t} - \mathcal{L}^*\mathbf{s}_0)/p$ for next base-p digits of $\mathbf{s}$
  (this reduces $\|\mathbf{t}'\|$ by p so finally can apply trivial algorithm)

Reduction from BDD to DGS + (Search) LWE

BDD *with bound* $\alpha p/\sqrt{2}r$ to DGS_r + search $\text{LWE}_{p,\alpha}$, for large enough r (in particular $r > \sqrt{2}p\,\eta_\varepsilon(\mathcal{L})$)

- Given $D_{\mathcal{L},r}$ samples (*instead of oracle*) and LWE oracle
- Want to solve BDD instance $(\mathcal{L}^*, \mathbf{t})$ (remark: the $\mathcal{L}^*$ part is always the same)
- Transform DG samples $\mathbf{v}$ to LWE samples $(\mathcal{L}^{-1}\mathbf{v} \bmod p, [\langle \mathbf{v}, \mathbf{t}\rangle/p + e] \bmod 1)$ where $e \sim D_{\alpha/\sqrt{2}}$
  - Suppose $\mathbf{v} = \mathcal{L}(\mathbf{a} + p\mathbf{a}')$, $\mathbf{a} \in \mathbb{Z}_p^n$, and $\mathbf{t} = \mathcal{L}^*\mathbf{s} + \mathbf{t}'$, $\mathcal{L}^*\mathbf{s}$ is CV
  - $\langle \mathcal{L}(p\mathbf{a}'), \mathcal{L}^*\mathbf{s}\rangle/p \bmod 1 = 0$
  - $\langle \mathbf{v}, \mathbf{t}'\rangle/p \overset{s}{\approx}$ Gaussian noise of width $r\|\mathbf{t}'\|/p \leq \alpha/\sqrt{2}$ (large enough r, "behaves like" continuous Gaussian!)
  - Samples $\overset{s}{\approx} (\mathbf{a}, [\langle \mathbf{a}, \mathbf{s}\rangle/p + e'] \bmod 1)$ where $e' \sim D_{\leq\alpha}$, $\mathbf{a} \overset{s}{\approx} U$
- Find $\mathbf{s}_0 = \mathbf{s} \bmod p$ using (enhanced) LWE oracle
- Recurse with $(\mathbf{t} - \mathcal{L}^*\mathbf{s}_0)/p$ for next base-p digits of $\mathbf{s}$ (this reduces $\|\mathbf{t}'\|$ by p so finally can apply trivial algorithm)

Reduction from BDD to DGS + (Search) LWE

BDD *with bound* $\alpha p/\sqrt{2}r$ to DGS_r + search $\mathrm{LWE}_{p,\alpha}$, for large enough r (in particular $r > \sqrt{2}p\,\eta_\varepsilon(\mathcal{L})$)

- Given $D_{\mathcal{L},r}$ samples (*instead of oracle*) and LWE oracle
- Want to solve BDD instance $(\mathcal{L}^*, \mathbf{t})$
  (remark: the $\mathcal{L}^*$ part is always the same)
- Transform DG samples $\mathbf{v}$ to LWE samples
  $(\mathcal{L}^{-1}\mathbf{v} \bmod p, [\langle \mathbf{v}, \mathbf{t}\rangle/p + e] \bmod 1)$ where $e \sim D_{\alpha/\sqrt{2}}$
  - Suppose $\mathbf{v} = \mathcal{L}(\mathbf{a} + p\mathbf{a}')$, $\mathbf{a} \in \mathbb{Z}_p^n$, and $\mathbf{t} = \mathcal{L}^*\mathbf{s} + \mathbf{t}'$, $\mathcal{L}^*\mathbf{s}$ is CV
  - $\langle \mathcal{L}(p\mathbf{a}'), \mathcal{L}^*\mathbf{s}\rangle/p \bmod 1 = 0$
  - $\langle \mathbf{v}, \mathbf{t}'\rangle/p \overset{s}{\approx}$ Gaussian noise of width $r\|\mathbf{t}'\|/p \le \alpha/\sqrt{2}$
    (large enough r, "behaves like" continuous Gaussian!)
  - Samples $\overset{s}{\approx} (\mathbf{a}, [\langle \mathbf{a}, \mathbf{s}\rangle/p + e'] \bmod 1)$ where $e' \sim D_{\le\alpha}$, $\mathbf{a} \overset{s}{\approx} U$
- Find $\mathbf{s}_0 = \mathbf{s} \bmod p$ using (enhanced) LWE oracle
- Recurse with $(\mathbf{t} - \mathcal{L}^*\mathbf{s}_0)/p$ for next base-p digits of $\mathbf{s}$
  (this reduces $\|\mathbf{t}'\|$ by p so finally can apply trivial algorithm)

Reduction from BDD to DGS + (Search) LWE

BDD *with bound* $\alpha p/\sqrt{2}r$ to DGS$_r$ + search LWE$_{p,\alpha}$, for large enough r (in particular $r > \sqrt{2}p\,\eta_\varepsilon(\mathcal{L})$)

- Given $D_{\mathcal{L},r}$ samples (*instead of oracle*) and LWE oracle
- Want to solve BDD instance $(\mathcal{L}^*, \mathbf{t})$
  (remark: the $\mathcal{L}^*$ part is always the same)
- Transform DG samples $\mathbf{v}$ to LWE samples
  $(\mathcal{L}^{-1}\mathbf{v} \bmod p, [\langle \mathbf{v}, \mathbf{t}\rangle/p + e] \bmod 1)$ where $e \sim D_{\alpha/\sqrt{2}}$
  - Suppose $\mathbf{v} = \mathcal{L}(\mathbf{a} + p\mathbf{a}')$, $\mathbf{a} \in \mathbb{Z}_p^n$, and $\mathbf{t} = \mathcal{L}^*\mathbf{s} + \mathbf{t}'$, $\mathcal{L}^*\mathbf{s}$ is CV
  - $\langle \mathcal{L}(p\mathbf{a}'), \mathcal{L}^*\mathbf{s}\rangle/p \bmod 1 = 0$
  - $\langle \mathbf{v}, \mathbf{t}'\rangle/p \overset{s}{\approx}$ Gaussian noise of width $r\|\mathbf{t}'\|/p \leq \alpha/\sqrt{2}$
    (large enough r, "behaves like" continuous Gaussian!)
  - Samples $\overset{s}{\approx} (\mathbf{a}, [\langle \mathbf{a}, \mathbf{s}\rangle/p + e'] \bmod 1)$ where $e' \sim D_{\leq\alpha}$, $\mathbf{a} \overset{s}{\approx} U$
- Find $\mathbf{s}_0 = \mathbf{s} \bmod p$ using (enhanced) LWE oracle
- Recurse with $(\mathbf{t} - \mathcal{L}^*\mathbf{s}_0)/p$ for next base-p digits of $\mathbf{s}$
  (this reduces $\|\mathbf{t}'\|$ by p so finally can apply trivial algorithm)

Reduction from BDD to DGS + (Search) LWE

BDD *with bound* $\alpha p/\sqrt{2}r$ to DGS_r + search $\mathsf{LWE}_{p,\alpha}$, for large enough r (in particular $r > \sqrt{2}p\,\eta_\varepsilon(\mathcal{L})$)

- Given $D_{\mathcal{L},r}$ samples (*instead of oracle*) and LWE oracle
- Want to solve BDD instance $(\mathcal{L}^*, \mathbf{t})$
  (remark: the $\mathcal{L}^*$ part is always the same)
- Transform DG samples $\mathbf{v}$ to LWE samples
  $(\mathcal{L}^{-1}\mathbf{v} \bmod p, [\langle \mathbf{v}, \mathbf{t}\rangle/p + e] \bmod 1)$ where $e \sim D_{\alpha/\sqrt{2}}$
  - Suppose $\mathbf{v} = \mathcal{L}(\mathbf{a} + p\mathbf{a}')$, $\mathbf{a} \in \mathbb{Z}_p^n$, and $\mathbf{t} = \mathcal{L}^*\mathbf{s} + \mathbf{t}'$, $\mathcal{L}^*\mathbf{s}$ is CV
  - $\langle \mathcal{L}(p\mathbf{a}'), \mathcal{L}^*\mathbf{s}\rangle/p \bmod 1 = 0$
  - $\langle \mathbf{v}, \mathbf{t}'\rangle/p \overset{s}{\approx}$ Gaussian noise of width $r\|\mathbf{t}'\|/p \leq \alpha/\sqrt{2}$
    (large enough r, "behaves like" continuous Gaussian!)
  - Samples $\overset{s}{\approx} (\mathbf{a}, [\langle \mathbf{a}, \mathbf{s}\rangle/p + e'] \bmod 1)$ where $e' \sim D_{\leq\alpha}$, $\mathbf{a} \overset{s}{\approx} U$
- Find $\mathbf{s}_0 = \mathbf{s} \bmod p$ using (enhanced) LWE oracle
- Recurse with $(\mathbf{t} - \mathcal{L}^*\mathbf{s}_0)/p$ for next base-p digits of $\mathbf{s}$
  (this reduces $\|\mathbf{t}'\|$ by p so finally can apply trivial algorithm)

Reduction from BDD to DGS + (Search) LWE

BDD *with bound* $\alpha p/\sqrt{2}r$ to DGS_r + search $\text{LWE}_{p,\alpha}$, for large enough r (in particular $r > \sqrt{2}p\,\eta_\varepsilon(\mathcal{L})$)

- Given $D_{\mathcal{L},r}$ samples (*instead of oracle*) and LWE oracle
- Want to solve BDD instance $(\mathcal{L}^*, \mathbf{t})$
  (remark: the $\mathcal{L}^*$ part is always the same)
- Transform DG samples $\mathbf{v}$ to LWE samples
  $(\mathcal{L}^{-1}\mathbf{v} \bmod p, [\langle \mathbf{v}, \mathbf{t}\rangle/p + e] \bmod 1)$ where $e \sim D_{\alpha/\sqrt{2}}$
  - Suppose $\mathbf{v} = \mathcal{L}(\mathbf{a} + p\mathbf{a}')$, $\mathbf{a} \in \mathbb{Z}_p^n$, and $\mathbf{t} = \mathcal{L}^*\mathbf{s} + \mathbf{t}'$, $\mathcal{L}^*\mathbf{s}$ is CV
  - $\langle \mathcal{L}(p\mathbf{a}'), \mathcal{L}^*\mathbf{s}\rangle/p \bmod 1 = 0$
  - $\langle \mathbf{v}, \mathbf{t}'\rangle/p \overset{s}{\approx}$ Gaussian noise of width $r\|\mathbf{t}'\|/p \le \alpha/\sqrt{2}$
    (large enough r, "behaves like" continuous Gaussian!)
  - Samples $\overset{s}{\approx} (\mathbf{a}, [\langle \mathbf{a}, \mathbf{s}\rangle/p + e'] \bmod 1)$ where $e' \sim D_{\le\alpha}$, $\mathbf{a} \overset{s}{\approx} U$
- Find $\mathbf{s}_0 = \mathbf{s} \bmod p$ using (enhanced) LWE oracle
- Recurse with $(\mathbf{t} - \mathcal{L}^*\mathbf{s}_0)/p$ for next base-p digits of $\mathbf{s}$
  (this reduces $\|\mathbf{t}'\|$ by p so finally can apply trivial algorithm)

Reduction from BDD to DGS + (Search) LWE

BDD *with bound* $\alpha p/\sqrt{2}r$ to DGS_r + search $\mathrm{LWE}_{p,\alpha}$, for large enough r (in particular $r > \sqrt{2}p\,\eta_\varepsilon(\mathcal{L})$)

- Given $D_{\mathcal{L},r}$ samples (*instead of oracle*) and LWE oracle
- Want to solve BDD instance $(\mathcal{L}^*, \mathbf{t})$
  (remark: the $\mathcal{L}^*$ part is always the same)
- Transform DG samples $\mathbf{v}$ to LWE samples
  $(\mathcal{L}^{-1}\mathbf{v} \bmod p, [\langle \mathbf{v}, \mathbf{t}\rangle/p + e] \bmod 1)$ where $e \sim D_{\alpha/\sqrt{2}}$
  - Suppose $\mathbf{v} = \mathcal{L}(\mathbf{a} + p\mathbf{a}')$, $\mathbf{a} \in \mathbb{Z}_p^n$, and $\mathbf{t} = \mathcal{L}^*\mathbf{s} + \mathbf{t}'$, $\mathcal{L}^*\mathbf{s}$ is CV
  - $\langle \mathcal{L}(p\mathbf{a}'), \mathcal{L}^*\mathbf{s}\rangle/p \bmod 1 = 0$
  - $\langle \mathbf{v}, \mathbf{t}'\rangle/p \overset{s}{\approx}$ Gaussian noise of width $r\|\mathbf{t}'\|/p \leq \alpha/\sqrt{2}$
    (large enough r, "behaves like" continuous Gaussian!)
  - Samples $\overset{s}{\approx} (\mathbf{a}, [\langle \mathbf{a}, \mathbf{s}\rangle/p + e'] \bmod 1)$ where $e' \sim D_{\leq\alpha}$, $\mathbf{a} \overset{s}{\approx} U$
- Find $\mathbf{s}_0 = \mathbf{s} \bmod p$ using (enhanced) LWE oracle
- Recurse with $(\mathbf{t} - \mathcal{L}^*\mathbf{s}_0)/p$ for next base-p digits of $\mathbf{s}$
  (this reduces $\|\mathbf{t}'\|$ by p so finally can apply trivial algorithm)

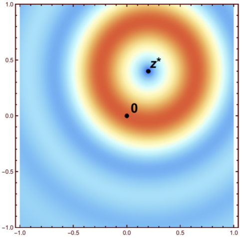
Reduction from BDD to DGS + (Search) LWE

BDD *with bound* $\alpha p/\sqrt{2}r$ to DGS_r + search $\text{LWE}_{p,\alpha}$, for large enough r (in particular $r > \sqrt{2}p\,\eta_\varepsilon(\mathcal{L})$)

- Given $D_{\mathcal{L},r}$ samples (*instead of oracle*) and LWE oracle
- Want to solve BDD instance $(\mathcal{L}^*, \mathbf{t})$
  (remark: the $\mathcal{L}^*$ part is always the same)
- Transform DG samples $\mathbf{v}$ to LWE samples
  $(\mathcal{L}^{-1}\mathbf{v} \bmod p, [\langle \mathbf{v}, \mathbf{t}\rangle/p + e] \bmod 1)$ where $e \sim D_{\alpha/\sqrt{2}}$
  - Suppose $\mathbf{v} = \mathcal{L}(\mathbf{a} + p\mathbf{a}')$, $\mathbf{a} \in \mathbb{Z}_p^n$, and $\mathbf{t} = \mathcal{L}^*\mathbf{s} + \mathbf{t}'$, $\mathcal{L}^*\mathbf{s}$ is CV
  - $\langle \mathcal{L}(p\mathbf{a}'), \mathcal{L}^*\mathbf{s}\rangle/p \bmod 1 = 0$
  - $\langle \mathbf{v}, \mathbf{t}'\rangle/p \overset{s}{\approx}$ Gaussian noise of width $r\|\mathbf{t}'\|/p \leq \alpha/\sqrt{2}$
    (large enough r, "behaves like" continuous Gaussian!)
  - Samples $\overset{s}{\approx} (\mathbf{a}, [\langle \mathbf{a}, \mathbf{s}\rangle/p + e'] \bmod 1)$ where $e' \sim D_{\leq\alpha}$, $\mathbf{a} \overset{s}{\approx} U$
- Find $\mathbf{s}_0 = \mathbf{s} \bmod p$ using (enhanced) LWE oracle
- Recurse with $(\mathbf{t} - \mathcal{L}^*\mathbf{s}_0)/p$ for next base-p digits of $\mathbf{s}$
  (this reduces $\|\mathbf{t}'\|$ by p so finally can apply trivial algorithm)

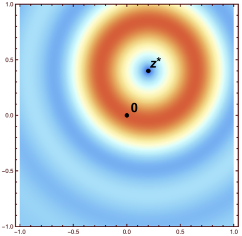
Reduction from BDD to DGS + Decisional LWE

- Transform samples in a similar way as in the search case, except with $\mathbf{t} \leftarrow \mathbf{t} - \mathbf{z}$ where $\mathbf{z}$ would be "guess" of $\mathbf{t}'$
- If $\mathbf{z}$ is good guess then get LWE samples of small error; otherwise get LWE samples of large error i.e. close to uniform
- Use generic algorithm for *oracle hidden center problem*
- (Requiring full flexibility of DGS: $D_{\mathcal{L},r_i}$ samples for $\{r_i \geq r\}_i$; fortunately the demand for r_i is still "static", regardless of $\mathbf{t}$)



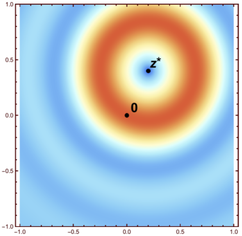
Reduction from BDD to DGS + Decisional LWE

- Transform samples in a similar way as in the search case, except with $\mathbf{t} \leftarrow \mathbf{t} - \mathbf{z}$ where $\mathbf{z}$ would be "guess" of $\mathbf{t}'$
- If $\mathbf{z}$ is good guess then get LWE samples of small error; otherwise get LWE samples of large error i.e. close to uniform
- Use generic algorithm for *oracle hidden center problem*
- (Requiring full flexibility of DGS: $D_{\mathcal{L},r_i}$ samples for $\{r_i \geq r\}_i$; fortunately the demand for r_i is still "static", regardless of $\mathbf{t}$)



Reduction from BDD to DGS + Decisional LWE

- Transform samples in a similar way as in the search case, except with $\mathbf{t} \leftarrow \mathbf{t} - \mathbf{z}$ where $\mathbf{z}$ would be "guess" of $\mathbf{t}'$
- If $\mathbf{z}$ is good guess then get LWE samples of small error; otherwise get LWE samples of large error i.e. close to uniform
- Use generic algorithm for *oracle hidden center problem*
- (Requiring full flexibility of DGS: $D_{\mathcal{L},r_i}$ samples for $\{r_i \geq r\}_i$; fortunately the demand for r_i is still "static", regardless of $\mathbf{t}$)



SKE from LWE

Scheme (encrypting one bit):

- Gen: $1^n \mapsto \mathbf{s}$ where $\mathbf{s} \sim \mathbb{Z}_p^n$
- Enc: $(\mathbf{s}, m) \mapsto (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e_p + m\lfloor p/2 \rfloor)$ where $\mathbf{a} \sim \mathbb{Z}_p^n$
- Dec: $(\mathbf{s}, c = (\mathbf{a}, b)) \mapsto [|b - \langle \mathbf{a}, \mathbf{s} \rangle| \geq p/4]$

Correctness: $|e_p| \leq p/4$ w.h.p.

Security: $\mathrm{Enc}(0) \overset{c}{\approx} U$ from LWE; then $\mathrm{Enc}(1) \overset{c}{\approx} U$ as well by adding $\lfloor p/2 \rfloor$; also multi-message as LWE supports multi-sample

Efficiency: key size $O(n \log p)$, message size $\times O(n \log p)$

SKE from LWE

Scheme (encrypting one bit):

- Gen: $1^n \mapsto \mathbf{s}$ where $\mathbf{s} \sim \mathbb{Z}_p^n$
- Enc: $(\mathbf{s}, m) \mapsto (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e_p + m\lfloor p/2 \rfloor)$ where $\mathbf{a} \sim \mathbb{Z}_p^n$
- Dec: $(\mathbf{s}, c = (\mathbf{a}, b)) \mapsto [|b - \langle \mathbf{a}, \mathbf{s} \rangle| \geq p/4]$

Correctness: $|e_p| \leq p/4$ w.h.p.

Security: $\text{Enc}(0) \overset{c}{\approx} U$ from LWE; then $\text{Enc}(1) \overset{c}{\approx} U$ as well by adding $\lfloor p/2 \rfloor$; also multi-message as LWE supports multi-sample

Efficiency: key size $O(n \log p)$, message size $\times O(n \log p)$

SKE from LWE

Scheme (encrypting one bit):

- Gen: $1^n \mapsto \mathbf{s}$ where $\mathbf{s} \sim \mathbb{Z}_p^n$
- Enc: $(\mathbf{s}, m) \mapsto (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e_p + m\lfloor p/2 \rfloor)$ where $\mathbf{a} \sim \mathbb{Z}_p^n$
- Dec: $(\mathbf{s}, c = (\mathbf{a}, b)) \mapsto [|b - \langle \mathbf{a}, \mathbf{s} \rangle| \geq p/4]$

Correctness: $|e_p| \leq p/4$ w.h.p.

Security: $\mathrm{Enc}(0) \overset{c}{\approx} U$ from LWE; then $\mathrm{Enc}(1) \overset{c}{\approx} U$ as well by adding $\lfloor p/2 \rfloor$; also multi-message as LWE supports multi-sample

Efficiency: key size $O(n \log p)$, message size $\times O(n \log p)$

SKE from LWE

Scheme (encrypting one bit):

- Gen: $1^n \mapsto \mathbf{s}$ where $\mathbf{s} \sim \mathbb{Z}_p^n$
- Enc: $(\mathbf{s}, m) \mapsto (\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e_p + m\lfloor p/2 \rfloor)$ where $\mathbf{a} \sim \mathbb{Z}_p^n$
- Dec: $(\mathbf{s}, c = (\mathbf{a}, b)) \mapsto [|b - \langle \mathbf{a}, \mathbf{s} \rangle| \geq p/4]$

Correctness: $|e_p| \leq p/4$ w.h.p.

Security: $\mathrm{Enc}(0) \stackrel{c}{\approx} U$ from LWE; then $\mathrm{Enc}(1) \stackrel{c}{\approx} U$ as well by adding $\lfloor p/2 \rfloor$; also multi-message as LWE supports multi-sample

Efficiency: key size $O(n \log p)$, message size $\times O(n \log p)$

PKE from LWE

Scheme (encrypting one bit):

- Gen: $1^n \mapsto (\mathbf{s}, \{(\mathbf{a}_i, b_i)\}_{i\in[k]})$ ($\mathbf{s}$ is the secret key) where $\mathbf{s}, \mathbf{a}_i \sim \mathbb{Z}_p^n$ and $b_i = \langle \mathbf{a}_i, \mathbf{s}\rangle + e_p^{(i)}$
- Enc: $(\{(\mathbf{a}_i, b_i)\}, m) \mapsto (\sum_{i\in S} \mathbf{a}_i, \sum_{i\in S} b_i + m\lfloor p/2\rfloor)$ where $S \sim \mathcal{P}([k])$
- Dec: $(\mathbf{s}, c = (\mathbf{a}, b)) \mapsto [|b - \langle \mathbf{a}, \mathbf{s}\rangle| \geq p/4]$

Correctness: $|ke_p| \leq p/4$ w.h.p.

Security: $(\{(\mathbf{a}_i, b_i)\}, \mathrm{Enc}(0)) \overset{c}{\approx} (U, \mathrm{Enc}_U(0))$ from LWE; $(U, \mathrm{Enc}_U(0)) \overset{s}{\approx} U$ for $k \geq (1+\delta)n\log p$; similar for Enc(1)

Efficiency: public key size $O(nk\log p)$, message size $\times O(n\log p)$

Potential optimization: $\mathbf{a}_i$ can be fixed in advance (while $e_p^{(i)}$ still need to be fresh), reducing public key size to $O(k\log p)$

PKE from LWE

Scheme (encrypting one bit):

- Gen: $1^n \mapsto (\mathbf{s}, \{(\mathbf{a}_i, b_i)\}_{i \in [k]})$ ($\mathbf{s}$ is the secret key) where $\mathbf{s}, \mathbf{a}_i \sim \mathbb{Z}_p^n$ and $b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_p^{(i)}$
- Enc: $(\{(\mathbf{a}_i, b_i)\}, m) \mapsto (\sum_{i \in S} \mathbf{a}_i, \sum_{i \in S} b_i + m\lfloor p/2 \rfloor)$ where $S \sim \mathcal{P}([k])$
- Dec: $(\mathbf{s}, c = (\mathbf{a}, b)) \mapsto [|b - \langle \mathbf{a}, \mathbf{s} \rangle| \geq p/4]$

Correctness: $|ke_p| \leq p/4$ w.h.p.

Security: $(\{(\mathbf{a}_i, b_i)\}, \mathrm{Enc}(0)) \overset{c}{\approx} (U, \mathrm{Enc}_U(0))$ from LWE; $(U, \mathrm{Enc}_U(0)) \overset{s}{\approx} U$ for $k \geq (1+\delta) n \log p$; similar for Enc(1)

Efficiency: public key size $O(nk \log p)$, message size $\times O(n \log p)$

Potential optimization: $\mathbf{a}_i$ can be fixed in advance (while $e_p^{(i)}$ still need to be fresh), reducing public key size to $O(k \log p)$

PKE from LWE

Scheme (encrypting one bit):

- Gen: $1^n \mapsto (\mathbf{s}, \{(\mathbf{a}_i, b_i)\}_{i\in[k]})$ ($\mathbf{s}$ is the secret key) where $\mathbf{s}, \mathbf{a}_i \sim \mathbb{Z}_p^n$ and $b_i = \langle \mathbf{a}_i, \mathbf{s}\rangle + e_p^{(i)}$
- Enc: $(\{(\mathbf{a}_i, b_i)\}, m) \mapsto (\sum_{i\in S} \mathbf{a}_i, \sum_{i\in S} b_i + m\lfloor p/2 \rfloor)$ where $S \sim \mathcal{P}([k])$
- Dec: $(\mathbf{s}, c = (\mathbf{a}, b)) \mapsto [|b - \langle \mathbf{a}, \mathbf{s}\rangle| \geq p/4]$

Correctness: $|ke_p| \leq p/4$ w.h.p.

Security: $(\{(\mathbf{a}_i, b_i)\}, \mathrm{Enc}(0)) \overset{c}{\approx} (U, \mathrm{Enc}_U(0))$ from LWE; $(U, \mathrm{Enc}_U(0)) \overset{s}{\approx} U$ for $k \geq (1+\delta)n \log p$; similar for Enc(1)

Efficiency: public key size $O(nk \log p)$, message size $\times O(n \log p)$

Potential optimization: $\mathbf{a}_i$ can be fixed in advance (while $e_p^{(i)}$ still need to be fresh), reducing public key size to $O(k \log p)$

PKE from LWE

Scheme (encrypting one bit):

- Gen: $1^n \mapsto (\mathbf{s}, \{(\mathbf{a}_i, b_i)\}_{i \in [k]})$ ($\mathbf{s}$ is the secret key) where $\mathbf{s}, \mathbf{a}_i \sim \mathbb{Z}_p^n$ and $b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_p^{(i)}$
- Enc: $(\{(\mathbf{a}_i, b_i)\}, m) \mapsto (\sum_{i \in S} \mathbf{a}_i, \sum_{i \in S} b_i + m\lfloor p/2 \rfloor)$ where $S \sim \mathcal{P}([k])$
- Dec: $(\mathbf{s}, c = (\mathbf{a}, b)) \mapsto [|b - \langle \mathbf{a}, \mathbf{s} \rangle| \geq p/4]$

Correctness: $|ke_p| \leq p/4$ w.h.p.

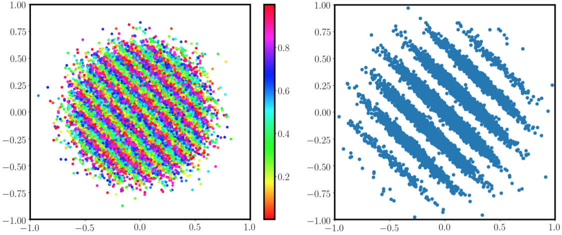
Security: $(\{(\mathbf{a}_i, b_i)\}, \mathrm{Enc}(0)) \overset{c}{\approx} (U, \mathrm{Enc}_U(0))$ from LWE; $(U, \mathrm{Enc}_U(0)) \overset{s}{\approx} U$ for $k \geq (1 + \delta) n \log p$; similar for Enc(1)

Efficiency: public key size $O(nk \log p)$, message size $\times O(n \log p)$

Potential optimization: $\mathbf{a}_i$ can be fixed in advance (while $e_p^{(i)}$ still need to be fresh), reducing public key size to $O(k \log p)$

Definition of (H)CLWE

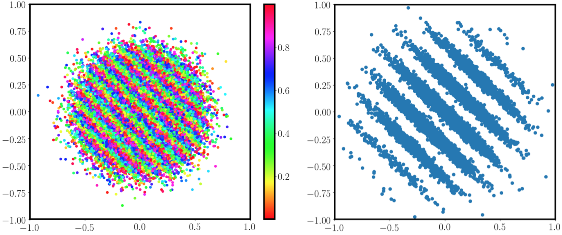
- Sample distribution $A_{\mathbf{w},\beta,\gamma}$: $(\mathbf{y}, z = [\gamma\langle \mathbf{y}, \mathbf{w}\rangle + e] \bmod 1)$ where $\mathbf{y} \sim D_1^n$ and $e \sim D_\beta$ (cf. $(\mathbf{a}, b = [\langle \mathbf{a}, \mathbf{s}, /\rangle p + e] \bmod 1)$, $\mathbf{a} \sim \mathbb{Z}_p^n$ and $e \sim D_\alpha$)
- Continuous LWE $\mathrm{CLWE}_{\beta,\gamma}$ (decision): Distinguish between $A_{\mathbf{w},\beta,\gamma}$ [3] and $D_1^n \times U([0,1))$
- Homogeneous variant $\mathrm{hCLWE}_{\beta,\gamma}$: Distinguish between $H_{\mathbf{w},\beta,\gamma}$ and D_1^n, where $H_{\mathbf{w},\beta,\gamma}$: $\mathbf{y} \mid (\mathbf{y}, z) \sim A_{\mathbf{w},\beta,\gamma}, z = 0$



[3] Average case: $\mathbf{w}$ is uniform unit vector

Definition of (H)CLWE

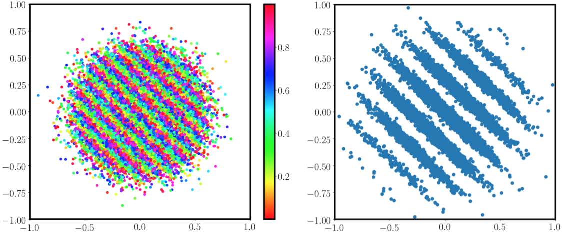
- Sample distribution $A_{\mathbf{w},\beta,\gamma}$: $(\mathbf{y}, z = [\gamma\langle \mathbf{y}, \mathbf{w}\rangle + e] \bmod 1)$ where $\mathbf{y} \sim D_1^n$ and $e \sim D_\beta$ (cf. $(\mathbf{a}, b = [\langle \mathbf{a}, \mathbf{s}, /\rangle p + e] \bmod 1)$, $\mathbf{a} \sim \mathbb{Z}_p^n$ and $e \sim D_\alpha$)
- Continuous LWE $\mathrm{CLWE}_{\beta,\gamma}$ (decision): Distinguish between $A_{\mathbf{w},\beta,\gamma}$ [3] and $D_1^n \times U([0,1))$
- Homogeneous variant $\mathrm{hCLWE}_{\beta,\gamma}$: Distinguish between $H_{\mathbf{w},\beta,\gamma}$ and D_1^n, where $H_{\mathbf{w},\beta,\gamma}$: $\mathbf{y} \mid (\mathbf{y}, z) \sim A_{\mathbf{w},\beta,\gamma}, z = 0$



[3] Average case: $\mathbf{w}$ is uniform unit vector

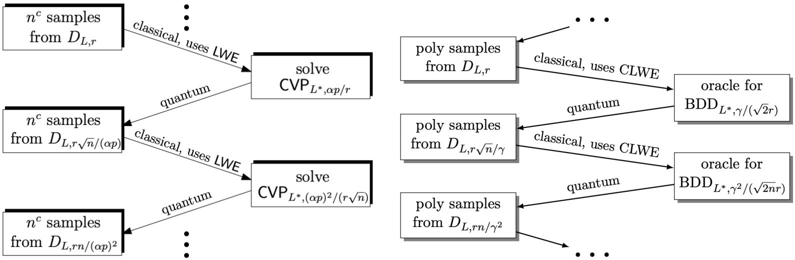
Definition of (H)CLWE

- Sample distribution $A_{\mathbf{w},\beta,\gamma}$: $(\mathbf{y}, z = [\gamma\langle \mathbf{y}, \mathbf{w}\rangle + e] \bmod 1)$ where $\mathbf{y} \sim D_1^n$ and $e \sim D_\beta$ (cf. $(\mathbf{a}, b = [\langle \mathbf{a}, \mathbf{s}, /\rangle p + e] \bmod 1)$, $\mathbf{a} \sim \mathbb{Z}_p^n$ and $e \sim D_\alpha$)
- Continuous LWE $\mathrm{CLWE}_{\beta,\gamma}$ (decision): Distinguish between $A_{\mathbf{w},\beta,\gamma}$ [3] and $D_1^n \times U([0,1))$
- Homogeneous variant $\mathrm{hCLWE}_{\beta,\gamma}$: Distinguish between $H_{\mathbf{w},\beta,\gamma}$ and D_1^n, where $H_{\mathbf{w},\beta,\gamma}$: $\mathbf{y} \mid (\mathbf{y}, z) \sim A_{\mathbf{w},\beta,\gamma}, z = 0$



[3] Average case: $\mathbf{w}$ is uniform unit vector

Reduction to CLWE: Preview



Reduction to (H)CLWE

Reduction from $\mathrm{DGS}_{2\sqrt{n}\eta_\varepsilon(\mathcal{L})/\beta}$ to $\mathrm{CLWE}_{\beta,\gamma}$ for $\gamma \geq 2\sqrt{n}$ and poly γ/β:

- Similar iterative reduction, reducing BDD to DGS + CLWE
- Transform to samples $((\mathbf{v}+\mathbf{e}_1)/R, [\langle \mathbf{v}, \mathbf{t}\rangle + e_2] \bmod 1)$ where $\mathbf{e}_1 \sim D_s^n$ and $e_2 \sim D_{\beta/\sqrt{2}}$, $s = \beta r/(\sqrt{2}\gamma)$, $R = \sqrt{r^2+s^2}$ (actually using r_i as oracle is decisional thus applying OHCP) (cf. $(\mathcal{L}^{-1}\mathbf{v} \bmod p, [\langle \mathbf{v}, \mathbf{t}\rangle/p + e] \bmod 1)$ where $e \sim D_{\alpha/\sqrt{2}}$)

Reduction from $\mathrm{CLWE}_{\beta,\gamma}$ to $\mathrm{hCLWE}_{\sqrt{\beta^2+\delta^2},\gamma}$ for poly $1/\delta$: rejection sampling on z with width δ

(Reduction from HCLWE to HCLWE with multiple hidden discrete directions: hybrid)

Reduction to (H)CLWE

Reduction from $\mathrm{DGS}_{2\sqrt{n}\eta_\varepsilon(\mathcal{L})/\beta}$ to $\mathrm{CLWE}_{\beta,\gamma}$ for $\gamma \geq 2\sqrt{n}$ and poly γ/β:

- Similar iterative reduction, reducing BDD to DGS + CLWE
- Transform to samples $((\mathbf{v}+\mathbf{e}_1)/R, [\langle \mathbf{v}, \mathbf{t}\rangle + e_2] \bmod 1)$ where $\mathbf{e}_1 \sim D_s^n$ and $e_2 \sim D_{\beta/\sqrt{2}}$, $s = \beta r/(\sqrt{2}\gamma)$, $R = \sqrt{r^2+s^2}$ (actually using r_i as oracle is decisional thus applying OHCP) (cf. $(\mathcal{L}^{-1}\mathbf{v} \bmod p, [\langle \mathbf{v}, \mathbf{t}\rangle/p + e] \bmod 1)$ where $e \sim D_{\alpha/\sqrt{2}}$)

Reduction from $\mathrm{CLWE}_{\beta,\gamma}$ to $\mathrm{hCLWE}_{\sqrt{\beta^2+\delta^2},\gamma}$ for poly $1/\delta$: rejection sampling on z with width δ

(Reduction from HCLWE to HCLWE with multiple hidden discrete directions: hybrid)

Reduction to (H)CLWE

Reduction from $\mathrm{DGS}_{2\sqrt{n}\eta_\varepsilon(\mathcal{L})/\beta}$ to $\mathrm{CLWE}_{\beta,\gamma}$ for $\gamma \geq 2\sqrt{n}$ and poly γ/β:

- Similar iterative reduction, reducing BDD to DGS + CLWE
- Transform to samples $((\mathbf{v} + \mathbf{e}_1)/R, [\langle \mathbf{v}, \mathbf{t} \rangle + e_2] \bmod 1)$ where $\mathbf{e}_1 \sim D_s^n$ and $e_2 \sim D_{\beta/\sqrt{2}}$, $s = \beta r/(\sqrt{2}\gamma)$, $R = \sqrt{r^2 + s^2}$ (actually using r_i as oracle is decisional thus applying OHCP) (cf. $(\mathcal{L}^{-1}\mathbf{v} \bmod p, [\langle \mathbf{v}, \mathbf{t} \rangle / p + e] \bmod 1)$ where $e \sim D_{\alpha/\sqrt{2}}$)

Reduction from $\mathrm{CLWE}_{\beta,\gamma}$ to $\mathrm{hCLWE}_{\sqrt{\beta^2+\delta^2},\gamma}$ for poly $1/\delta$: rejection sampling on z with width δ

(Reduction from HCLWE to HCLWE with multiple hidden discrete directions: hybrid)

Reduction to (H)CLWE

Reduction from $\mathrm{DGS}_{2\sqrt{n}\eta_\varepsilon(\mathcal{L})/\beta}$ to $\mathrm{CLWE}_{\beta,\gamma}$ for $\gamma \geq 2\sqrt{n}$ and poly γ/β:

- Similar iterative reduction, reducing BDD to DGS + CLWE
- Transform to samples $((\mathbf{v}+\mathbf{e}_1)/R, [\langle \mathbf{v}, \mathbf{t}\rangle + e_2] \bmod 1)$ where $\mathbf{e}_1 \sim D_s^n$ and $e_2 \sim D_{\beta/\sqrt{2}}$, $s = \beta r/(\sqrt{2}\gamma)$, $R = \sqrt{r^2+s^2}$ (actually using r_i as oracle is decisional thus applying OHCP) (cf. $(\mathcal{L}^{-1}\mathbf{v} \bmod p, [\langle \mathbf{v}, \mathbf{t}\rangle/p + e] \bmod 1)$ where $e \sim D_{\alpha/\sqrt{2}}$)

Reduction from $\mathrm{CLWE}_{\beta,\gamma}$ to $\mathrm{hCLWE}_{\sqrt{\beta^2+\delta^2},\gamma}$ for poly $1/\delta$: rejection sampling on z with width δ

(Reduction from HCLWE to HCLWE with multiple hidden discrete directions: hybrid)

Misc. about (H)CLWE

- (H)CLWE with $\beta = 0$ ("noiseless") can be solved by LLL
- HCLWE can be solved by checking the eigenvalues of the covariance matrix estimated from $2^{O(\gamma^2)}$ samples
- HCLWE gives hardness of estimating Gaussian mixtures
- Besides lattice-based hardness, HCLWE also enjoys concrete *statistical query* hardness

References

[Reg09]: reduction to (search) LWE, reductions among LWE variants, PKE from LWE
[PRSD17]: reduction to decisional LWE
[BRST21]: (H)CLWE

Joan Bruna, Oded Regev, Min Jae Song, and Yi Tang.
Continuous LWE.
In *STOC'21—Proceedings of the 53th Annual ACM SIGACT Symposium on Theory of Computing*, pages 694–707. ACM, New York, 2021.

Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz.
Pseudorandomness of ring-LWE for any ring and modulus.
In *STOC'17—Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 461–473. ACM, New York, 2017.

Oded Regev.
On lattices, learning with errors, random linear codes, and cryptography.
J. ACM, 56(6):Art. 34, 40, 2009.
Preliminary version in STOC 2005.